

## A Comparative Study On Hadoop Ecosystem: Hive And HBase – A Literature Review

Moh Rifqi Zamzami<sup>1</sup>, Moh Riswandha Imawan<sup>2</sup>, Imam Ghozali<sup>3</sup>.

Universitas Pembangunan Nasional Veteran Jawa Timur<sup>1,3</sup>

Universitas Muhammadiyah Surabaya<sup>2</sup>

\*Korespondensi: [rifkiinstag@gmail.com](mailto:rifkiinstag@gmail.com)

### Abstract

*In the current era, where handling vast volumes of data is a pivotal challenge for businesses, this research conducts a systematic literature review comparing Hive and HBase within the Hadoop ecosystem. The study filters relevant articles from diverse sources, including accredited SINTA national journals, international journals, and Scopus-accredited journals. Results deepen understanding of the roles and contributions of Hive and HBase components in processing big data, aiding companies in selecting the most suitable solution for their data processing and security needs. The research highlights that HBase excels in fast, random read/write operations, while Hive is more efficient in data querying. However, both tools' performance is influenced by factors like data size, node quantity, and system configuration. This study provides a profound comprehension of Hive and HBase in addressing big data challenges, laying the groundwork for further development. Intended for IT practitioners and researchers, this research contributes valuable insights into big data processing within the Hadoop.*

### Keywords:

*HBase, Hive, Hadoop, Big Data*

### Abstrak

Dalam era di mana volume data yang besar semakin menjadi tantangan utama bagi perusahaan, penelitian ini bertujuan untuk melakukan tinjauan literatur yang komprehensif terkait dengan perbandingan antara Hive dan HBase dalam ekosistem Hadoop. Dengan menggunakan pendekatan systematic literature review, penelitian ini mengumpulkan artikel-artikel terpilih dari berbagai sumber, termasuk jurnal nasional terakreditasi SINTA, jurnal internasional, dan jurnal akreditasi Scopus. Berhasil disaring artikel-artikel yang relevan dengan fokus pembahasan seperti arsitektur, proses pengolahan data, dan penerapan HBase dan Hive. Hasil dari penelitian ini memberikan pemahaman yang lebih mendalam tentang peran dan kontribusi masing-masing komponen dalam memproses big data pada ekosistem Hadoop. Dengan memahami bagaimana HBase dan Hive memproses data serta peran masing-masing dalam mengamankan informasi, perusahaan dapat memilih solusi yang paling cocok untuk kebutuhan pengolahan dan keamanan data mereka. Penelitian ini juga membandingkan kinerja dari kedua alat tersebut. Hasilnya menunjukkan bahwa HBase memiliki kinerja yang lebih baik dalam operasi read/write yang cepat dan acak, sedangkan Hive lebih efisien dalam melakukan query data. Namun, kinerja dari kedua alat ini juga dipengaruhi oleh faktor-faktor seperti ukuran data, jumlah node, dan konfigurasi sistem. Penelitian ini memberikan pemahaman mendalam tentang peran dan kontribusi HBase dan Hive dalam mengatasi tantangan pemrosesan big data, serta memberikan landasan bagi pengembangan lebih lanjut dalam hal ini. Dengan demikian, penelitian ini diharapkan dapat memberikan wawasan yang berharga bagi para praktisi dan peneliti di bidang teknologi informasi, khususnya dalam konteks pengolahan big data menggunakan ekosistem Hadoop.

### Kata kunci:

*HBase, Hive, Hadoop, Big Data*

## PENDAHULUAN

Memastikan bahwa volume data yang besar dapat disimpan, dimanipulasi, dan dipulihkan dengan efisien adalah salah satu masalah teknis utama yang dihadapi oleh perusahaan saat ini. Layanan online dan media sosial adalah penyebab utama mengapa alat pengolahan data diciptakan sekarang ini. Pengguna Facebook mengunggah 450 miliar foto dan video setiap hari dan menghasilkan 4 juta like pada setiap menit [1]. Setiap hari, aplikasi media sosial, internet of things (IoT), data geografis, *vector space*, daya listrik, sensor nirkabel, dan jaringan listrik menghasilkan volume data yang jauh lebih besar dari petabyte [1][2]. Data ini mungkin mengandung informasi penting yang terkadang tidak dapat diolah dengan baik oleh teknologi saat ini.

Ketika jumlah data terus meningkat setiap hari, ketidakefisienan dalam penyimpanan dan pemrosesan data menimbulkan masalah yang sistemik. Hal ini disebabkan karena tidak ada alat dan metode konvensional yang dibuat untuk menangani dan mengelola kompleksitas big data [3][4]. Karena volume, variasi, dan nilai data yang terus meningkat, metode dan alat konvensional tidak dapat mengatasi masalah tersebut. Penting untuk mengelola jumlah data yang besar ini dan masalah signifikan yang menyertainya dengan efisien. Merancang sebuah skema yang sesuai dengan masalah big data saat ini adalah tugas yang penting bagi para *database designer*.

Setiap hari, perusahaan-perusahaan besar di bidang transportasi, peramalam cuaca seperti BMKG, Internet of Things (IoT), energi, serta media sosial seperti Facebook, Instagram, Twitter, hingga TikTok, menghasilkan petabyte data dengan berbagai jenis dan format. Twitter memproses sekitar 400 miliar peristiwa secara real-time dan menghasilkan data dalam skala petabyte setiap harinya. Sementara itu, Instagram mengumpulkan berbagai data tentang penggunanya, termasuk konten yang diposting, komentar dan likes, juga informasi seperti situs web yang dikunjungi, aplikasi yang digunakan, atau pembelian hingga pembelian yang dilakukan oleh pengguna. Kemudian menurut BBC, TikTok mengumpulkan sejumlah besar data pengguna termasuk video yang ditonton dan dikomentari, data lokasi, model ponsel dan sistem operasi yang digunakan, ritme pengetikan orang saat mengetik, serta kontak dan pesan [5][6][7]. Data ini digunakan untuk pengambilan keputusan di masa depan, di mana keputusan tersebut pasti memiliki pengaruh yang signifikan terhadap bagaimana perusahaan dapat bertahan nantinya. Metode tradisional mungkin tidak dapat menyimpan dan mengelola berbagai format dan jenis data yang ada dalam dataset ini. Salah satu masalah yang muncul saat menangani dataset yang besar adalah memastikan penyimpanan yang memadai, bukan hanya memproses atau menganalisis data.

Terdapat alat-alat khusus yang dikembangkan seiring berjalannya waktu untuk menyimpan dan memproses big data secara akurat. Hadoop adalah salah satu alat yang memiliki kemampuan untuk memproses big data yang terstruktur, tidak terstruktur, dan semi-terstruktur. Hadoop memiliki dua paradigma utama: Hadoop Storage Area dan Hadoop Distributed File System (HDFS). Selain itu, Hadoop map-reduce adalah alat pemrosesan yang kuat yang dapat digunakan untuk memproses data yang disimpan. Di dalam ekosistem Hadoop, terdapat beberapa alat khusus yang dapat memproses, menyimpan, dan menganalisis big data, diantaranya adalah HBase dan Hive. HBase adalah alat pemrosesan data besar lain yang dikembangkan di atas Hadoop dan dijalankan di atas Hadoop

Distributed File System (HDFS). Karena arsitektur HDFS yang kaku, ia tidak dapat berubah sesuai dengan parameter dataset. Facebook pertama kali memilih HBase untuk mengimplementasikan platform pesan baru pada November 2010 [8]. Pada tahun 2010, Apache HBase dimulai sebagai proyek oleh perusahaan powerset karena kebutuhan untuk memproses jumlah data pencarian bahasa alami yang besar. Sementara itu, sama seperti HBase, Hive adalah alat pemrosesan big data lainnya yang dikembangkan di atas Hadoop dan beroperasi di atas Hadoop Distributed File System (HDFS). Berbeda dengan HBase yang dirancang untuk bekerja dengan model data wide-column store, Hive menawarkan antarmuka SQL yang memungkinkan pengguna menyusun kueri menggunakan bahasa SQL yang kompleks.

HBase dan Hive adalah dua alat pemrosesan big data yang populer yang dikembangkan di atas Hadoop. Keduanya memiliki arsitektur pemrosesan data yang berbeda. HBase menggunakan arsitektur model data wide-column store, sedangkan Hive menggunakan arsitektur model data relational. Arsitektur model data wide-column store pada HBase memungkinkannya untuk menyimpan data yang terus diperbarui secara real-time. Hal ini karena HBase tidak memerlukan proses reorganisasi data secara berkala. Namun, arsitektur ini juga memiliki kelemahan, yaitu dapat menyebabkan overhead yang besar untuk data yang jarang diperbarui. Sementara arsitektur model data relational pada Hive memungkinkannya untuk menyimpan data yang terstruktur dengan efisien. Hal ini karena Hive menggunakan bahasa SQL yang sudah dikenal luas. Namun, arsitektur ini juga memiliki kelemahan, yaitu tidak dapat menyimpan data yang terus diperbarui secara real-time.

Fokus utama dalam penelitian ini adalah untuk membandingkan arsitektur pemrosesan data pada HBase dan Hive. Penelitian ini akan membahas secara mendalam kelebihan dan kekurangan dari masing-masing arsitektur. Penelitian ini juga akan membandingkan kinerja dari kedua alat tersebut.

### *1.1 Apache HBase*

HBase adalah basis data NoSQL open source dengan laten rendah dan throughput tinggi [8]. HBase memanfaatkan infrastruktur Hadoop, seperti HDFS dan Zookeeper, dan terkenal karena operasi read/write yang cepat dan acak, menjadikannya pilihan populer untuk menangani big data dengan pemrosesan yang cepat. Dengan kemajuan dalam arsitektur dinamisnya, kerangka kerja HBase menawarkan metode untuk mengekstraksi, memanipulasi, dan menyimpan big data [4]. Ini mendukung integritas, skalabilitas, dan ketahanan kesalahan data. Penggunaan Apache HBase untuk menyimpan dan mengelola dataset dengan mengubah arsitektur penyimpanan sesuai dengan parameter dataset telah mendorong popularitas Apache Hadoop di dunia akademik dan bisnis.

### *1.2 Apache Hive*

Hive adalah sebuah perangkat lunak data warehouse yang diberdayakan oleh Apache dan bersifat open source. Dirancang untuk mengelola dan melakukan kueri terhadap kumpulan data yang tersebar, Hive menggunakan SQL sebagai bahasa utamanya. Dalam konteks ini, Hive membantu dalam membentuk konsep data yang terstruktur, memfasilitasi manipulasi dan analisis data secara efisien [9]. Hive menyediakan bahasa kueri bernama

HiveQL, yang mirip dengan SQL, memungkinkan pengguna untuk mengakses dan mengolah data dengan familiaritas SQL. Metastore merupakan komponen kunci yang berfungsi sebagai penyimpanan metadata dalam arsitektur Hive dan berjalan di JVM yang sama dengan layanan Hive. Untuk menyimpan metadata, Hive menggunakan database tertanam bernama Derby, yang diimplementasikan dalam sistem file lokal dalam mode tertentu. Metastore Hive yang efektif memastikan manajemen metadata yang baik dan membantu pencarian dan pengelolaan data yang lebih efisien dalam lingkungan data yang didistribusikan.

## TINJAUAN PUSTAKA

### 2.1.1 Pemilihan Database

Dalam metodologi penelitian ini, perencanaan tinjauan pustaka dilakukan dengan mempertimbangkan sumber-sumber yang memiliki kredibilitas dan kualitas yang tinggi. Sumber-sumber tersebut mencakup jurnal akreditasi SINTA, jurnal internasional, dan jurnal akreditasi Scopus. Pemilihan ketiga jenis jurnal ini didasarkan pada keinginan untuk mengakses literatur yang telah melalui proses seleksi yang ketat dan memiliki reputasi yang baik dalam dunia ilmiah. Jurnal akreditasi SINTA memberikan akses terhadap publikasi yang telah diakui oleh lembaga akreditasi nasional, sementara jurnal internasional dan akreditasi Scopus memberikan jangkauan yang lebih luas ke dalam literatur internasional dan terindeks secara internasional. Pemilihan database ini diharapkan dapat memberikan dasar yang solid untuk analisis dan sintesis literatur dalam kerangka tinjauan pustaka penelitian ini.

### 2.1.2 Menentukan Kata Kunci

Langkah selanjutnya dalam metodologi adalah menentukan kata kunci atau keyword untuk pencarian artikel yang relevan setelah pemilihan database. Kata kunci yang dipilih untuk penelitian ini mencakup aspek penting dari penelitian, seperti "HBase", "Hive", "Hadoop", "Database", "Big Data", "HDFS", dan "Data Processing." Pemilihan kata kunci ini didasarkan pada kebutuhan untuk menemukan penelitian yang secara khusus membahas aspek yang terkait dengan HBase dan Hive dalam pengolahan big data.

### 2.1.3 Menentukan Kriteria Inklusi dan Eksklusi

Untuk merinci tinjauan pustaka dalam penelitian ini, langkah pertama yang sangat penting adalah menentukan standar inklusi dan eksklusi literatur. Hal ini dilakukan untuk memastikan bahwa literatur yang dipilih tidak hanya relevan, tetapi juga secara berkualitas sesuai dalam fokus penelitian.

Tabel 1. Inklusi dan Eksklusi

No	Inclusion Rules	Exclusion Rules
1	Literatur dari jurnal akreditasi SINTA, jurnal internasional, dan jurnal akreditasi Scopus	Literatur yang tidak relevan dengan fokus penelitian, misalnya yang hanya menyentuh Hadoop tanpa fokus khusus pada HBase dan Hive

---

2	Secara khusus membahas arsitektur pemrosesan data dan penerapan dari HBase atau Hive	Literatur yang tidak terbit pada jurnal ilmiah
3	Literatur terbitan 7 tahun terakhir, yakni minimal tahun 2017	Literatur yang tidak membahas aspek kinerja dan penerapan dari HBase atau Hive

---

## 2.2 Peninjauan Artikel Terpilih

Pada tahap awal peninjauan artikel, sebanyak 33 artikel berhasil dikumpulkan dari berbagai database termasuk SINTA, Jurnal Internasional, dan Scopus. Untuk menyaring artikel-artikel tersebut, peneliti melakukan pembacaan judul, abstrak, dan kesimpulan dari setiap artikel yang relevan terkait fokus pembahasan seperti arsitektur, proses pengolahan data, dan penerapan HBase dan Hive. Proses membaca ini dilakukan untuk mengidentifikasi artikel yang memiliki relevansi tinggi dengan pembahasan yang berkaitan. Setelah melalui tahap tersebut, peneliti berhasil menyaring artikel menjadi 20 artikel yang memenuhi kriteria penelitian yang telah ditetapkan.

## 2.3 Perumusan Pertanyaan Penelitian

Penting untuk menetapkan pertanyaan utama yang akan membantu literature review pada penelitian ini. Pertanyaan-pertanyaan tersebut mencakup aspek penting seperti perbandingan arsitektur, evaluasi kinerja, dan integrasi antara HBase dan Hive. Oleh karena itu, tujuan utama dari penelitian ini adalah untuk menjelaskan temuan terkini yang dapat membantu memahami fungsi dan kontribusi masing-masing komponen dalam memproses big data pada ekosistem Hadoop. Berikut adalah pertanyaannya.

Tabel 2. Pertanyaan Penelitian

<b>RQ1</b>	Bagaimana arsitektur pemrosesan data pada HBase dan Hive?
<b>RQ2</b>	Bagaimana perbandingan kinerja antara HBase dan Hive dalam pengolahan big data?
<b>RQ3</b>	Bagaimana arsitektur yang dimiliki HBase dan Hive dapat mengamankan data

## METODE

Metodologi ataupun metode yang diadaptasi dalam penelitian ini adalah menggunakan pendekatan *systematic literature review*. Dalam melakukan penelitian didasarkan pada beberapa jurnal yang informasinya berkaitan dengan pembahasan arsitektur serta penerapan HBase dan Hive pada studi kasus tertentu. Jurnal yang didapatkan bersumber dari jurnal nasional terakreditasi minimal SINTA dan jurnal internasional.

Systematic literature review bertujuan untuk mencari, menemukan dan mensintesis artikel atau literatur secara sistematis terkait dengan studi atau penelitian sebelumnya dengan proses yang terorganisir dengan baik dan transparan, yang menggunakan prosedur dalam setiap langkahnya. Selain itu, systematic literature review juga mengacu pada proses

identifikasi, evaluasi, serta penafsiran semua penelitian yang tersedia dan relevan dengan pertanyaan, area topik atau fenomena menarik dari penelitian yang dilakukan oleh peneliti.

Tujuan dari systematic literature review adalah untuk mencari, menemukan, dan mensintesis artikel atau literatur yang secara sistematis terkait dengan penelitian atau penelitian sebelumnya melalui proses yang terorganisir dengan baik dan transparan, yang menggunakan prosedur dalam setiap langkahnya. Systematic literature review juga mengacu pada proses identifikasi, evaluasi, dan penafsiran semua penelitian yang saat ini tersedia dan relevan dengan pertanyaan, area topik, atau fenomena yang dibahas dalam penelitian tersebut [10].

## HASIL DAN PEMBAHASAN

Dalam upaya untuk merinci dan menelaah literatur dengan lebih terfokus, peneliti menyusun Tabel 3 berikut yang memuat daftar artikel terpilih. Artikel-artikel ini dipilih berdasarkan kriteria yang mencakup relevansi topik penelitian. Tabel berikut menjadi landasan untuk analisis literatur yang mendalam dan sintesis yang akan peneliti bahas lebih lanjut.

Tabel 3. Daftar Artikel Terpilih

Judul	Penulis	Tahun Publikasi	Jurnal Publikasi
Evaluation of NoSQL Databases: MongoDB, Cassandra, HBase, ...[11]	Houcine Matallah, Ghalem Belalem	2020	International Journal of Software Science and Computational Intelligence
Efficient data replay mechanism of sensor stream data ...[12]	Feng Ye, Jun Sun, Zeyin Du, Nadia Nedjah, Wei Liu, Lin Lan	2022	Journal of King Saud University – Computer and Information Sciences (Scopus)
Databases fit for blockchain technology... [13]	Jovan Kalajdjieski, Mayank Raikwar, Nino Arsov, Goran Velinov, Danilo Gligoroski	2023	Blockchain: Research and Applications (Scopus)
MDICA: Maintenance of data integrity in column-oriented...[14]	María Jos´e Suarez-Cabal, Pablo Suarez-Otero, Claudio de la Riva , Javier Tuya	2023	Computer Standards & Interfaces (Scopus)
Research on School Intelligent Classroom Management...[15]	Zhi Min Zhu, Fang Qin Xua, Xia Gao	2020	Procedia Computer Science (Scopus)
Research of Temporal Information Index Strategy Based on HBase... [16]	Chao Feng, Baoan Li	2017	Procedia Computer Science (Scopus)
Analysis of NoSQL Schema Design approaches using HBase for GIS...[17]	Monika Sharmaa, Mahesh Bundele	2019	Procedia Computer Science (Scopus)
Introduction and Performance: An Overview of Hive...[18]	Li Ding, Li Hsin Cheng	2017	Department of Computer Science: Rochester University
Uji Performa dan Perbandingan RDBMS Mysql dan Hive-Hadoop...[9]	Nurul Azizah, Henry Saptono	2020	Jurnal Informatika Terpadu (SINTA 5)
Using Distributed Data over HBase in Big Data...[19]	Dillon Chrimes1, Hamid Zamani	2017	Hindawi – Computational and Mathematical Methods in Medicine

Relational Databases Versus HBase: An Experimental Evaluation...[20]	Zakaria Bousalem, Inssaf El Guabassi, Ilias Cherti	2019	ASTES (Advances in Science, Technology and Engineering Systems)
Performance Comparison Between Apache Hive and...[21]	Rotsnarani Sethy, Santosh Kumar Dash, Mrutyunjaya Panda	2017	Proceedings of the Eighth International Conference on Soft Computing and Pattern Recognition
Big Data Analytics Using Hadoop Tools – Apache Hive...[22]	Prof R.Angelin Preethi, Prof J.Elavarasi	2017	International Journal of Emerging Technology in Computer Science & Electronics (IJETCSE)
Data Processing in Hive vs. SQL Server: A comparative analysis...[23]	Nadeem Ahmed, Shakil Ahamed, Jahir Ibna Rafiq, Sifatur Rahim	2019	IEEE International Conference on Engineering Technologies and Social Sciences (ICETSS)
An Overview of Apache Pig and Apache Hive[24]	Saiyam Arora, Abinesh Verma, Richa Vasuja	2019	International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT)
Perbandingan Kinerja HBase dan MongoDB Sebagai Backend IoT...[25]	Niki Yuniar Wicaksono, Eko Sakti Pramukantoro, Widhi Yahya	2018	Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer (SINTA 3)
Perbandingan Performa Database Apache HBase dan Apache Cassandra Sebagai...[26]	Dimas Malik Ibrahim, Rakhmadhany, Primananda, Mahendra Data	2018	Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer (SINTA 3)
Study and Improvement of Performance of NoSQL Databases...[27]	Bila Khonde Noell, Boluma Mangata Bopatriciat, Mbuyi Mukendi Eugène, Bukanga Christian Parfum	2022	IJISCS (International Journal of Information System and Computer Science)
Hive: A Literature Review [28]	Kavita D. Mahajan, Vijay D. Chaudhari	2019	International Journal of Innovations in Engineering and Science
Predictive analytics on COVID-19 data using Hive based on Hadoop cluster [29]	Ali Abbood Khaleel, Ali Noori Kareem, Laith Hikmet Mahdi	2023	Indonesian Journal of Electrical Engineering and Computer Science (SINTA 1 & Scopus)

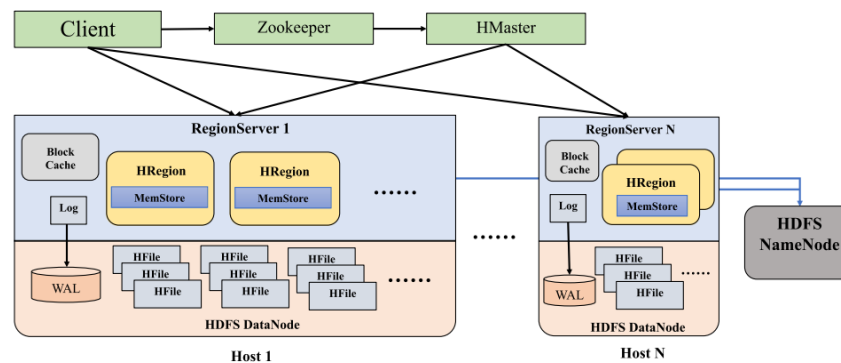
Dengan artikel-artikel terpilih yang telah disusun dalam Tabel 3 sebagai landasan. Selanjutnya peneliti akan melihat bagaimana HBase dan Hive memproses data, bagaimana mereka dapat digunakan, dan peran masing-masing dalam mengamankan informasi. Memahami bagian-bagian ini sangatlah penting untuk memahami kemampuan teknis keduanya, serta untuk menilai seberapa cocok mereka untuk pengolahan big data dan keamanan data.

### 3.1 Arsitektur

Untuk menjawab pertanyaan penelitian yang pertama (RQ1), dilkakukan analisis pada artikel terpilih terkait dengan bagaimana arsitektur HBase dan Hive dibangun guna melakukan pengolahan pada big data. Dengan memahami arsitektur ini, peneliti dapat menggali lebih dalam tentang cara HBase dan Hive merancang fondasi mereka untuk menangani tantangan pemrosesan data berskala besar. Berikut adalah penjelasan rincinya.

### 3.1.1 Arsitektur HBase

Arsitektur HBase terdiri dari tiga komponen utama seperti pada Gambar 1, yang mana bekerja sama untuk menyediakan penyimpanan dan pengelolaan data yang terdistribusi yaitu RegionServers, HBase Master, dan Zookeeper [30].



Gambar 1. Arsitektur HBase

#### 1. RegionServers

Sebagai bagian penting dari struktur HBase, RegionServers bertanggung jawab atas penyimpanan dan pengelolaan data dalam kluster. Mereka berinteraksi langsung dengan klien, menanggapi permintaan data terkait, dan mengelola wilayah data, yang merupakan bagian dari tabel yang disimpan. Selain itu, RegionServers mengatur operasi baca dan tulis pada data di dalamnya, menjadikannya bagian penting dari pemrosesan HBase.

#### 2. HBase Master

HBase Master, sebagai manajer utama, memiliki tugas kunci dalam mengelola penugasan wilayah data ke RegionServers. Selain itu, HBase Master menangani manajemen tabel secara keseluruhan, menyediakan layanan tingkat tinggi seperti penanganan kesalahan dan pemulihan, serta memastikan distribusi merata wilayah data di seluruh kluster. Dengan fungsi-fungsi utamanya, HBase Master menjadi penentu keseimbangan dan ketersediaan sistem.

#### 3. Zookeeper

Sebagai penyelar dan pemantau, Zookeeper memainkan peran penting dalam menjaga konsistensi dan koordinasi kluster HBase. Dia bertanggung jawab untuk memantau status seluruh komponen, memilih master, dan memastikan bahwa kluster HBase secara keseluruhan aman dan berfungsi dengan baik.

#### 4. Operasi Dasar

HBase, sebagai database NoSQL berbasis kolom, mendukung operasi baca/tulis sederhana. Operasi Get digunakan untuk mencari data spesifik berdasarkan kunci, sedangkan operasi Scan memungkinkan pemindaian data massal dengan kriteria tertentu. Operasi Put menambahkan atau memperbarui data, sedangkan operasi Delete menghapus data. Keempat operasi ini membentuk dasar interaksi dengan data di dalam HBase.

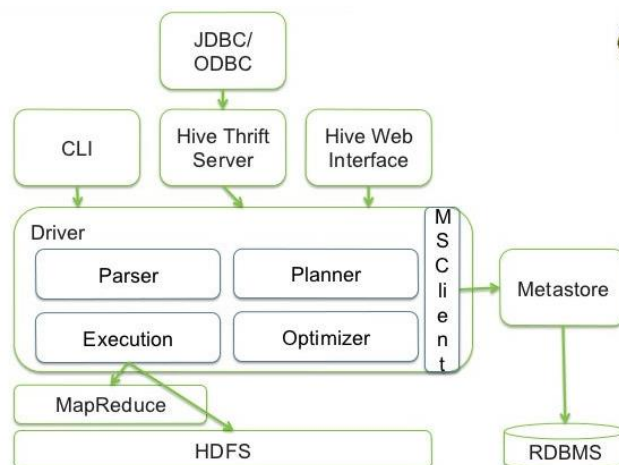
#### 5. Distribusi dan Skalabilitas

Skalabilitas horizontal HBase memungkinkan kluster mengatasi beban data yang lebih besar dengan menambah sumber daya secara efisien, memberikan fleksibilitas penting dalam menghadapi tuntutan data besar. Selain itu, struktur HBase memungkinkan penambahan RegionServers secara dinamis untuk memastikan distribusi beban kerja yang merata, sementara HBase Master menjaga keseimbangan wilayah data di antara RegionServers.

Arsitektur pada HBase memberikan keuntungan yang signifikan dalam pengolahan big data, meliputi distribusi yang merata pada data, kinerja baca/tulis yang tinggi, efisiensi pemrosesan data kolom, dan tingkat ketersediaan yang tinggi. Keunggulan-keunggulan ini menjadikan HBase sebagai salah satu pilihan utama untuk pengolahan big data.

### 3.1.2 Arsitektur Hive

Arsitektur dari Apache Hive terdiri dari beberapa komponen seperti metastore, compiler, optimizer, executor, UI, CLI, dan thrift server (client) [22].



Gambar 2. Arsitektur Hive

1. Metastore  
Komponen ini bertanggung jawab untuk menyimpan metadata terkait tabel, skema, lokasi data, dan metadata partisi. Metadata ini penting untuk melacak kemajuan data yang tersebar di cluster. Metastore menggunakan tradisional RDBMS untuk menyimpan metadata, dan backup server secara teratur mereplikasi data untuk mengantisipasi kehilangan data.
2. Compiler  
Compiler melakukan kompilasi kueri HiveQL, mengonversi kueri ke rencana eksekusi. Proses ini melibatkan konversi kueri ke pohon sintaksis abstrak (AST) dan kemudian ke grafik acyclic directed (DAG). DAG membagi operator ke tahap MapReduce dan tugas berdasarkan kueri masukan dan data .
3. Optimizer  
Optimizer menerima rencana eksekusi yang dihasilkan oleh compiler dan melakukan berbagai optimisasi logis, seperti proyeksi dan pushdown. Tujuannya adalah untuk meningkatkan kinerja eksekusi kueri dengan mengoptimalkan rencana eksekusi .
4. Executor  
Setelah proses kompilasi dan optimisasi, Executor bertanggung jawab untuk mengeksekusi tugas sesuai dengan rencana eksekusi yang dihasilkan. Ini berinteraksi dengan job tracker Hadoop untuk menjadwalkan tugas yang akan dijalankan dan memastikan bahwa tugas-tugas tersebut dieksekusi dengan benar .
5. CLI, UI, AND THRIFT SERVER  
Antarmuka Baris Perintah (CLI) dan Antarmuka Pengguna (UI) memungkinkan pengguna eksternal berinteraksi dengan Hive dengan mengirimkan kueri, instruksi, dan memantau status proses. Server Thrift memungkinkan klien eksternal berinteraksi dengan Hive seperti halnya JDBC/ODBC.

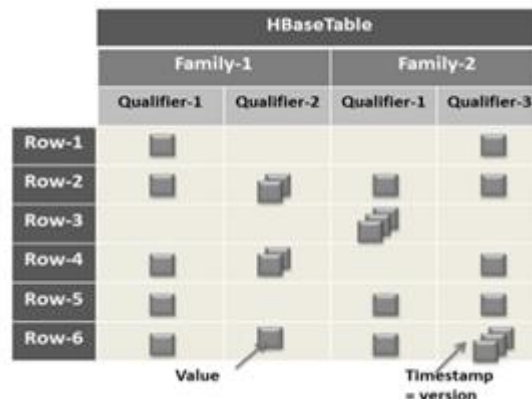
Hive mengadopsi arsitektur MapReduce sebagai landasan eksekusi query dalam lingkungannya. Arsitektur MapReduce, yang digunakan oleh Hive, terbukti menjadi pendekatan yang sangat efisien dalam memproses volumi data yang besar.

**3.2 Pemrosesan Data**

Masih menjawab pertanyaan penelitian pertama (RQ1), pembahasan berikut akan melanjutkan fokus dari arsitektur ke pendalaman bagaimana HBase dan Hive memproses data.

**3.2.1 Data Model pada HBase**

Seperti yang sudah disebutkan, HBase termasuk dalam kategori database Column-Oriented dan didesain untuk memberikan akses real-time ke data yang disimpan di HDFS. HBase juga didasarkan pada model BigTable yang dikembangkan oleh Google. Konsep-konsep penting dalam HBase adalah bagian integral dari cara HBase menyimpan dan mengakses data [20]. Berikut adalah penjelasan komprehensif tentang konsep-konsep tersebut.

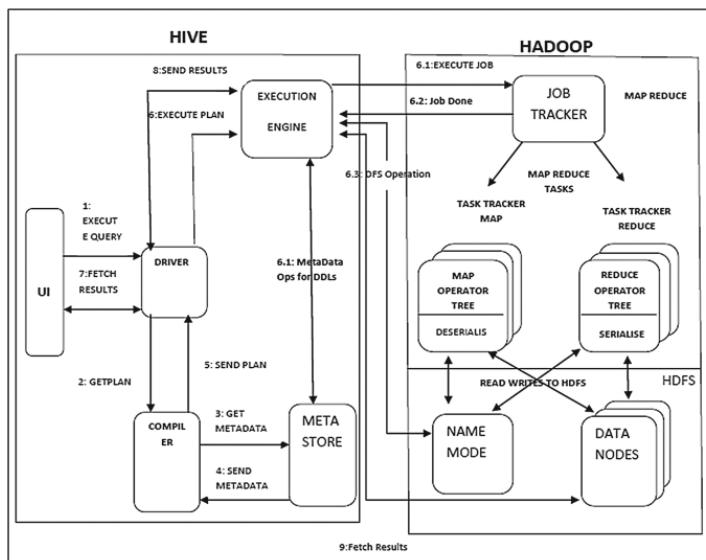


Gambar 3. Model Pemrosesan Data Pada HBase

RowKey, kunci khusus yang digunakan untuk mengakses baris tertentu, diidentifikasi untuk setiap baris dalam tabel HBase. Karena data diurutkan berdasarkan RowKey, pemilihan RowKey yang tepat dapat memengaruhi kinerja akses. Dalam HBase, beberapa versi dari setiap Cell yang diidentifikasi oleh Timestamp, membantu HBase mendukung beberapa versi dari data tersebut. Column Families ditetapkan saat pembuatan tabel dan tidak dapat diubah; Column Families berisi satu atau lebih Column Qualifiers yang digunakan untuk mengakses data di dalamnya. Cell, sebagai unit penyimpanan HBase, terdiri dari RowKey, Column Family, Column Qualifier, dan Timestamp.

**3.2.2 Alur Pemrosesan pada Hive**

Dalam Apache Hive, terdapat sembilan tahapan dalam pemrosesan query pada Hadoop seperti pada Gambar 4 [21].



Gambar 4. Alur Pemrosesan Data Pada Hive

Tahap pertama adalah User Interface yang memanggil execute query ke Driver. Kemudian, pada tahap kedua, Driver membuat session handle untuk query dan mengirimkannya ke Compiler untuk menghasilkan rencana eksekusi untuk Execution Engine. Pada tahap ketiga dan keempat, Compiler mendapatkan metadata yang diperlukan dari Metastore. Pada tahap kelima, Compiler mengirimkan rencana yang dihasilkan ke Execution Engine. Seluruh komponen Apache Hive

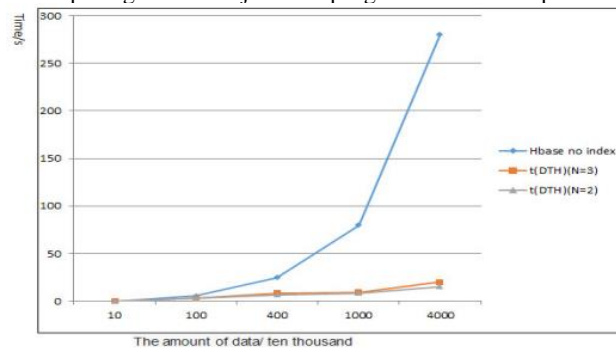
terhubung dengan model HDFS pada tahap 6.1, 6.2, dan 6.3. Pada tahap 7, 8, dan 9, Apache Hive melakukan operasi mapper dan reducer.

### 3.3 Kinerja dalam Penerapannya

Untuk menjawab pertanyaan penelitian yang kedua (RQ2), peneliti menelaah artikel terpilih yang memberikan pembahasan mengenai bagaimana kinerja HBase dan Hive dalam penerapan terhadap studi kasus dari masing-masing artikel. Berikut penjelasan rincinya.

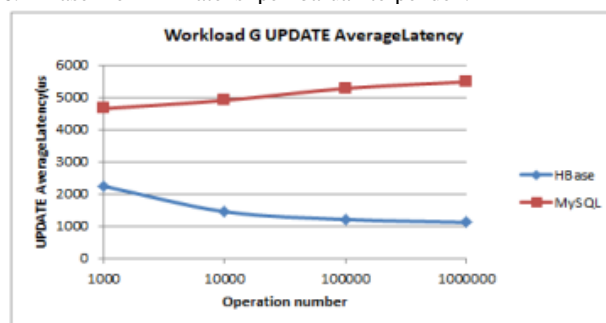
#### 3.3.1 Performa HBase

Dalam artikel [16], dalam penerapannya untuk penyimpanan dan pengambilan data temporal, HBase mempunyai keunggulan dalam hal efisiensi dan kecepatan. Penggunaan HBase untuk menyimpan data medis elektronik temporal dalam skala besar telah terbukti memberikan peningkatan kinerja dalam pengambilan data temporal.



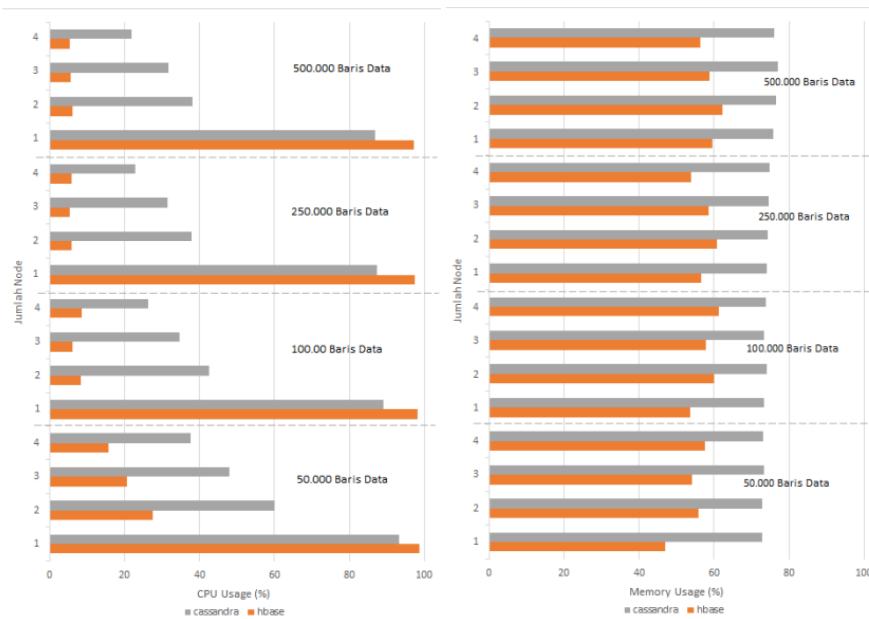
Gambar 5. Uji Temporal Data

Kemudian artikel [20] mengemukakan bahwa HBase menunjukkan kinerja yang lebih baik dalam operasi I/O bound (menulis data) tetapi kinerja yang lebih buruk dalam operasi CPU bound (membaca data) jika dibandingkan dengan MySQL. Namun, HBase menunjukkan kinerja yang lebih baik dalam memuat data dibandingkan dengan MySQL. HBase menunjukkan penurunan latensi yang besar, berbeda dengan MySQL yang menunjukkan peningkatan yang stabil seperti yang ditunjukkan pada Gambar 6. HBase memiliki latensi pembaruan terpendek.



Gambar 6. Workload G Update Latency

Kemudian pada artikel [26], dimana dilakukan perbandingan kinerja antara HBase dan Cassandra yang menunjukkan HBase memiliki keunggulan dalam penggunaan CPU Usage dan Memory Usage yang lebih baik daripada Cassandra dalam memasukkan data seperti pada Gambar 7. Hal ini disebabkan karena HBase menggunakan resource memori yang lebih sedikit dibandingkan dengan Cassandra, meskipun menjalankan jumlah aplikasi yang lebih banyak seperti Hadoop, Zookeeper, dan HBase itu sendiri.

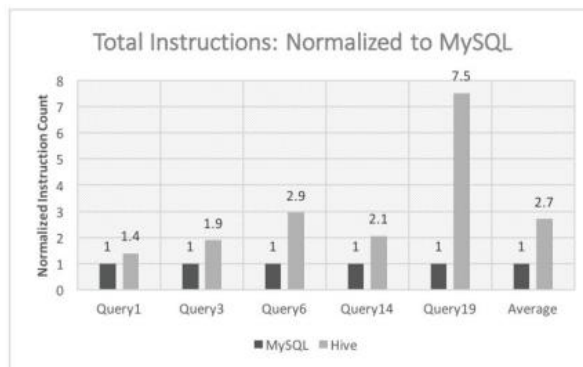


Gambar 7. Memory and CPU Usage

Kemudian dalam artikel [27], performa HBase tergantung pada jenis operasi yang dilakukan pada data. Dalam workload D yang terdiri dari 95% operasi baca dan 5% operasi penambahan data, HBase menunjukkan performa yang buruk dengan waktu eksekusi sekitar 43 detik. Namun, dalam workload E yang terdiri dari 95% operasi pemindaian cepat dan 5% operasi penambahan data, HBase menunjukkan performa yang sangat baik dengan waktu eksekusi hanya 1 menit 15 detik. Dalam workload F yang terdiri dari 50% operasi baca dan 50% operasi modifikasi tulis.

### 3.3.2 Performa Hive

Dalam artikel [18], dengan lebih dari 7500 workload yang dikirimkan ke kluster dan pemrosesan lebih dari 75TB data terkompresi setiap hari, Hive memiliki kemampuan untuk menangani volume data yang mencapai petabyte, seperti yang ditunjukkan oleh penggunaannya di Facebook. Kemudian, Hive menonjol dalam kemampuan kueri dengan menawarkan bahasa deklaratif yang serupa dengan SQL, yang kemudian dikompilasi menjadi pekerjaan map-reduce yang dieksekusi secara efektif menggunakan Hadoop. Basis penggunaannya yang luas memungkinkan pengguna baru untuk beradaptasi dengan cepat. Terdapat perbandingan kinerja Hive dan MySQL, yang mencakup perbedaan dalam waktu inisialisasi, periode pemanasan, dan kinerja keseluruhan pada Gambar 8 berikut.



Gambar 8. Performa Query Keseluruhan Antara Hive dan MySQL

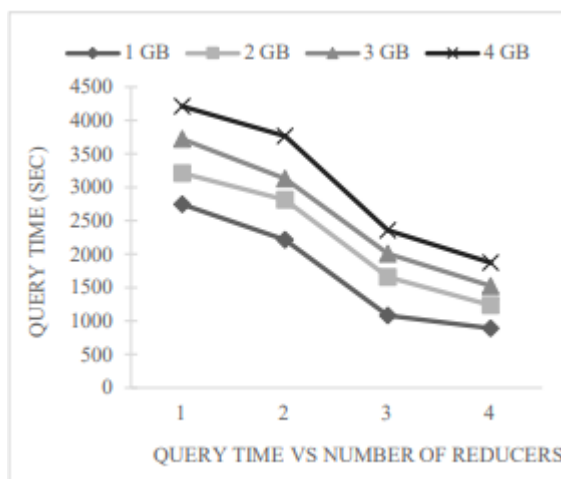
Kemudian pada artikel [9], performa Hive dievaluasi melalui pengujian yang melibatkan pemasangan Hadoop, Hive, dan Derby, serta langkah-langkah seperti pembuatan database, tabel, pengisian data, dan eksekusi query select. Waktu yang diperlukan untuk setiap operasi dicatat dalam mengevaluasi performa Hive pada pengolahan big data. Hasil pengujian memberikan gambaran efisiensi dan kemampuan Hive dalam menangani pengisian data dan query select pada skala dataset yang besar seperti pada Tabel 4.

Tabel 4. Perbandingan Waktu Eksekusi Kueri

Ukuran Data	Select Data MySQL	Select Data Hive
1 GB	40, 2 sec	0 sec, 8 map

2 GB	50, 3 sec	0 sec, 17 map
3 GB	59, 02 sec	0 sec, 26 map
4 GB	62, 34 sec	0 sec, 34 map
5 GB	67, 97 sec	0 sec, 43 map

Selanjutnya, dalam sebuah penelitian yang menggunakan Hive untuk menganalisis data COVID-19, hasilnya menunjukkan bahwa kinerja Hive dalam hal waktu pemrosesan kueri dan skalabilitas melebihi sistem manajemen basis data relasional (RDBMS) [29].



Gambar 9. Eksekusi Kueri Menggunakan Beberapa Reducer

Hive, sebagai platform analisis data, memiliki skalabilitas tinggi dan kinerja yang optimal. Dirancang untuk beroperasi di kluster Hadoop dengan menggunakan sejumlah mesin komoditas, Hive memungkinkan analisis, rangkuman, dan kueri terhadap besar volume data yang tersimpan di dalam Hadoop Distributed File System (HDFS). Dengan pendekatan skalabilitas horizontal, Hive dapat dengan mudah meningkatkan kapasitas dan kinerjanya dengan menambahkan lebih banyak mesin ke dalam kluster. Meskipun menggunakan model pemrosesan MapReduce, Hive tetap menghadirkan kecepatan eksekusi kueri yang signifikan dengan mengimplementasikan optimisasi dan teknik-teknik seperti partisi dan indeks. Dengan demikian, Hive memberikan solusi efektif untuk mengatasi tuntutan pemrosesan data skala besar dengan skala dan kecepatan yang seimbang.

### 3.4 Keamanan Data

Untuk menjawab pertanyaan penelitian ketiga (RQ3) terkait keamanan yang dipunyai oleh HBase dan Hive dalam memproteksi data berskala besar, peneliti kembali menyaring artikel terpilih yang memuat informasi terkait.

Berdasarkan artikel [18], beberapa fitur yang ditawarkan oleh Hive seperti; autentikasi sebagai langkah awal dalam memastikan bahwa hanya pengguna yang sah yang dapat mengakses data di dalam Hive. Otorisasi, sebagai lanjutan logis dari autentikasi, mengontrol ketat hak pengguna untuk mengakses data dan objek di dalamnya, sehingga meningkatkan keamanan dan privasi. Selain itu, Hive sangat baik dalam mendukung enkripsi data, yang membuatnya menjadi lapisan perlindungan tambahan yang baik untuk mencegah orang yang tidak berhak mengakses data yang disimpan. Dengan menggabungkan autentikasi, otorisasi, dan enkripsi data, Hive memberikan fondasi keamanan yang kokoh untuk menjaga kerahasiaan dan integritas data dalam lingkungan pengolahan data yang luas.

Kemudian artikel [19] yang berfokus pada penerapan HBase di dalam platform pelayanan kesehatan, menyebutkan bahwa HBase memainkan peran penting dalam menjamin keamanan data pasien. Pertama, HBase memungkinkan penyimpanan aman dengan menyimpan data pasien yang terenkripsi sesuai dengan peraturan privasi dan keamanan data kesehatan seperti HIPAA, FIPPA, dan PIPA. Selanjutnya, HBase memastikan bahwa tidak ada orang yang dapat mengakses data pasien, memenuhi persyaratan keamanan data yang ketat dalam konteks yang lebih khusus. Kemudian, melalui kemampuan replikasi data, HBase memungkinkan penggunaan algoritma masking atau enkripsi yang diperlukan untuk menjaga keamanan data sensitif, menjadikan platform ini tidak hanya aman tetapi juga sesuai dengan standar keamanan tertinggi dalam analisis data kesehatan.

## **PENUTUP**

### **Kesimpulan**

Dari analisis mendalam terhadap literatur mengenai arsitektur dan performa pemrosesan big data menggunakan HBase dan Hive, dapat ditarik beberapa kesimpulan utama. HBase, yang didasarkan pada arsitektur distribusi model BigTable, unggul dalam aspek keandalan dan kemampuan skalabilitas horizontalnya. Kelebihan HBase tampak jelas terutama dalam operasi I/O bound dan pembaruan data dengan tingkat latensi yang rendah. Sebaliknya, Hive menawarkan pendekatan yang berbeda dengan memanfaatkan antarmuka SQL untuk mempermudah penggunaan dan menangani volume data besar melalui model pemrosesan MapReduce.

Namun terlepas dari perbedaan tersebut, penting untuk diingat bahwa masing-masing dari keduanya memiliki keunggulan dan kelemahan, dan keputusan penggunaan antara HBase atau Hive harus didasarkan pada kebutuhan proyek tertentu. Sementara Hive lebih cocok untuk skenario yang berfokus pada kueri dan analisis data yang lebih kompleks, HBase lebih cocok untuk aplikasi yang membutuhkan latensi rendah dan pembaruan data yang cepat.

### **Saran**

Beberapa saran dapat dipertimbangkan dalam rangka pengembangan yang lebih lanjut. Pertama, penelitian lebih lanjut diperlukan untuk meningkatkan kinerja HBase dalam operasi CPU bound. Kedua, perlu dieksplorasi integrasi lebih lanjut antara HBase dan Hive untuk meningkatkan sinergi kedua platform. Ketiga, fokus pada aspek keamanan data pada HBase dan Hive untuk memastikan integritas dan kerahasiaan data. Keempat, pengembangan model data yang dapat diintegrasikan dengan baik oleh kedua platform untuk mendukung pemrosesan data yang lebih efisien. Terakhir, identifikasi kasus penggunaan terbaik untuk HBase dan Hive berdasarkan karakteristik data dan kebutuhan bisnis untuk memberikan panduan praktis bagi pengguna implementasi.

## **DAFTAR PUSTAKA**

- [1] D. Reinsel, J. Gantz, and J. Rydning, "The Digitization of the World From Edge to Core," Framingham, MA 01701, Nov. 2018. Accessed: Dec. 26, 2023. [Online]. Available: <https://www.seagate.com/files/www-content/our-story/trends/files/idc-seagate-dataage-whitepaper>
- [2] J. Leonard, "19 Data and Analytics Predictions Through 2025," Business2Community. Accessed: Dec. 23, 2023. [Online]. Available: <https://www.business2community.com/big-data/19-data-and-analytics-predictions-through-2025-02178668>
- [3] J. Zhang, G. Wu, X. Hu, and X. Wu, "A distributed cache for Hadoop Distributed File System in real-time cloud services," in *Proceedings - IEEE/ACM International Workshop on Grid Computing*, 2012, pp. 12–21. doi: 10.1109/Grid.2012.17.
- [4] M. U. Hassan, I. Yaqoob, S. Zulfiqar, and I. A. Hameed, "A comprehensive study of HBase storage architecture-a systematic literature review," *Symmetry (Basel)*, vol. 13, no. 1, pp. 1–21, Jan. 2021, doi: 10.3390/sym13010109.
- [5] Mint Fox, "How much data does TikTok use?," Mint Mobile. Accessed: Dec. 23, 2023. [Online]. Available: <https://www.mintmobile.com/blog/how-much-data-does-tiktok-use/>
- [6] L. Zhang and C. Malife, "Processing billions of events in real time at Twitter," Twitter Blog. Accessed: Dec. 23, 2023. [Online]. Available: [https://blog.twitter.com/engineering/en\\_us/topics/infrastructure/2021/processing-billions-of-events-in-real-time-at-twitter-](https://blog.twitter.com/engineering/en_us/topics/infrastructure/2021/processing-billions-of-events-in-real-time-at-twitter-)
- [7] J. Tidy, "TikTok: What is the app and how much data does it collect?," BBC. Accessed: Dec. 23, 2023. [Online]. Available: <https://www.bbc.com/news/technology-53476117>
- [8] L. George, *HBase: The Definitive Guide: Random Access to Your Planet-Size Data*, First Edition. Sebastopol, CA, USA: O'Reilly Media, Inc, 2011.
- [9] N. Azizah and H. Saptono, "UJI PERFORMA DAN PERBANDINGAN RDBMS MYSQL DAN HIVE-HADOOP," *Jurnal Informatika Terpadu*, vol. 6, no. 1, pp. 20–28, Mar. 2020, [Online]. Available: <https://journal.nurulfikri.ac.id/index.php/JIT>
- [10] W. B. Alfajri, A. Puji Widodo, and K. Adi, "Penerapan Tata Kelola Teknologi Informasi pada Instansi: Systematic Literature Review," *Jurnal Nasional Teknologi dan Sistem Informasi*, vol. 7, no. 3, pp. 191–198, Jan. 2022, doi: 10.25077/teknosi.v7i3.2021.191-198.
- [11] H. Matallah, G. Belalem, and K. Bouamrane, "Evaluation of NoSQL Databases," *International Journal of Software Science and Computational Intelligence*, vol. 12, no. 4, pp. 71–91, Sep. 2020, doi: 10.4018/ijssci.2020100105.
- [12] F. Ye, J. Sun, Z. Du, N. Nedjah, W. Liu, and L. Lan, "Efficient data replay mechanism of sensor stream data based on concurrent buffer pool," *Journal of King Saud University - Computer and Information Sciences*, vol. 34, no. 10, pp. 10293–10303, Nov. 2022, doi: 10.1016/j.jksuci.2022.10.021.
- [13] J. Kalajdjeski, M. Raikwar, N. Arsov, G. Velinov, and D. Gligoroski, "Databases fit for blockchain technology: A complete overview," *Blockchain: Research and Applications*, vol. 4, no. 1. Zhejiang University, Mar. 01, 2023. doi: 10.1016/j.bcr.2022.100116.
- [14] M. J. Suárez-Cabal, P. Suárez-Otero, C. de la Riva, and J. Tuya, "MDICA: Maintenance of data integrity in column-oriented database applications," *Comput Stand Interfaces*, vol. 83, Jan. 2023, doi: 10.1016/j.csi.2022.103642.
- [15] Z. M. Zhu, F. Q. Xu, and X. Gao, "Research on school intelligent classroom management system based on internet of things," in *Procedia Computer Science*, Elsevier B.V., 2020, pp. 144–149. doi: 10.1016/j.procs.2020.02.037.
- [16] C. Feng and B. Li, "Research of Temporal Information Index Strategy Based on HBase," in *Procedia Computer Science*, Elsevier B.V., 2017, pp. 367–372. doi: 10.1016/j.procs.2017.03.119.
- [17] M. Sharma and M. Bundele, "Analysis of NoSQL schema design approaches using HBase for GIS data," in *Procedia Computer Science*, Elsevier B.V., 2019, pp. 59–65. doi: 10.1016/j.procs.2019.05.027.
- [18] L. Ding and L. Hsin Cheng, "Introduction and Performance: An Overview of Hive," New York, USA, 2017.
- [19] D. Chrimes and H. Zamani, "Using Distributed Data over HBase in Big Data Analytics Platform for Clinical Services," *Comput Math Methods Med*, vol. 2017, 2017, doi: 10.1155/2017/6120820.
- [20] Z. Bousalem, I. El Guabassi, and I. Cherti, "Relational databases versus HBase: An experimental evaluation," *Advances in Science, Technology and Engineering Systems*, vol. 4, no. 2, pp. 395–401, 2019, doi: 10.25046/aj040249.
- [21] R. Sethy, S. K. Dash, and M. Panda, "Performance comparison between apache hive and oracle SQL for big data analytics," in *Advances in Intelligent Systems and Computing*, Springer Verlag, 2018, pp. 130–141. doi: 10.1007/978-3-319-60618-7\_14.
- [22] A. Preetih and J. Elavarasi, "BIG DATA ANALYTICS USING HADOOP TOOLS – APACHE HIVE VS APACHE PIG," *International Journal of Emerging Technology in Computer Science & Electronics (IJETCSE)*, vol. 24, no. 3, Feb. 2017.
- [23] N. Ahmed, S. Ahamed, J. I. Rafiq, and S. Rahim, "Data processing in Hive vs. SQL server: A comparative analysis in the query performance," in *2017 IEEE 3rd International Conference on Engineering Technologies and Social Sciences, ICETSS 2017*, Institute of Electrical and Electronics Engineers Inc., Jul. 2017, pp. 1–5. doi: 10.1109/ICETSS.2017.8324202.
- [24] S. Arora, A. Verma, R. Vasuja, and R. Vasuja, "An Overview of Apache Pig and Apache Hive," *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, pp. 432–436, Mar. 2019, doi: 10.32628/cseit195250.
- [25] N. Y. Wicaksono, E. Sakti Pramukantoro, and W. Yahya, "Perbandingan Kinerja HBase dan MongoDB Sebagai Backend IoT Data Storage," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 2, no. 12, pp. 6842–6848, 2018, [Online]. Available: <http://j-ptiik.ub.ac.id>
- [26] D. Malik Ibrahim, R. Primananda, and M. Data, "Perbandingan Performa Database Apache HBase dan Apache Cassandra Sebagai Media Penyimpanan Data Sensor Internet of Things," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 2, no. 8, pp. 2943–2949, Aug. 2018, [Online]. Available: <http://j-ptiik.ub.ac.id>
- [27] B. KHONDE Noel, B. MANGATA Bopatriciat, M. MUKENDI Eugène, and B. CHRISTIAN Parfum, "STUDY AND IMPROVEMENT OF PERFORMANCE OF NoSQL DATABASES: MongoDB, HBase and OrientDB," *IJISCS (International Journal of Information System and Computer Science)*, pp. 164–172, 2022, [Online]. Available: <http://www.oracle.com/technetwork/java/ja>
- [28] K. D. Mahajan and V. D. Chaudhari, "Hive: A Literature Review," *International Journal of Innovations in Engineering and Science*, vol. 4, no. 10, p. 425003, 2019, [Online]. Available: [www.ijies.net](http://www.ijies.net)
- [29] A. A. Khaleel, A. N. Kareem, and L. H. Mahdi, "Predictive analytics on COVID-19 data using Hive based on Hadoop cluster," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 31, no. 2, pp. 945–956, Aug. 2023, doi: 10.11591/ijeecs.v31.i2.pp945-956.
- [30] Z. Cao, H. Dong, Y. Wei, S. Liu, and D. H. C. Du, "IS-HBase: An In-Storage Computing Optimized HBase with I/O Offloading and Self-Adaptive Caching in Compute-Storage Disaggregated Infrastructure," *ACM Transactions on Storage*, vol. 18, no. 2, May 2022, doi: 10.1145/3488368.

- ZamZami, M. R., Wibowo, N. C., Ana Wati, S. F., Ghozali, I., & Imawan, M. R. (2024). Rancang Bangun Sistem Informasi Berbasis Web Menggunakan Metode Waterfall. *CYCLOTRON*, 7(01), 61–66. <https://doi.org/10.30651/cl.v7i01.21084>
- Ghozali, I., Riswandha Imawan, M., Rifqi Zamzami, M., Zuhri, S., Pembangunan Nasional Veteran Jawa Timur, U., & Muhammadiyah Surabaya, U. (2023). WEBMAP UNTUK PENGEMBANGAN JALUR IRIGASI BARU DI KABUPATEN LAMONGAN. 1(5). <https://doi.org/10.47353/satukata.v1i5.1401>
- Riswandha Imawan, M., Rifqi Zamzami, M., Ghozali, I., Muhammadiyah Surabaya, U., & Pembangunan Nasional Veteran Jawa Timur, U. (2023). PANDANGAN ORANG TUA DALAM PENGGUNAAN APLIKASI MEDIA SOSIAL DI ANAK REMAJA (STUDI KASUS: KOTA SURABAYA). 1(4). <https://doi.org/10.47353/satukata.v1i4.1015>
- Riswandha Imawan, M. (2023). MODEL PEMBELAJARAN KOOPERATIF TIPE TPS-TGT PADA PEMBELAJARAN MATEMATIKADI KELAS VIII SMP N 1 SEMARANG. 1(1), 1–9. <https://doi.org/10.3342/jursih.v1i1.14>
- Riswanda, M., & Ghozali, I. (2020). Tips & Trick Android Root:Cara Cepat dan Mudah Belajar Tips & Trick Android. Jakad Media Publishing. [www.nandroid19.com](http://www.nandroid19.com)